Justin P. Peters

SLIS S603-Agent Based Modeling Workshop

Summer I 2011

Final Project Reflections

For my project in Agent Based Modeling S603 I decided to investigate the Agent Analyst extension for the ArcGIS software suite. Given my background in GIS software, I was excited to integrate this new (to me anyway) modeling technique into the graphic and visualization tools available within ArcGIS.

After receiving the download from the assistant instructor, Ali Ghazinejad, I worked through the documentation on installation and setting up the system variables to launch the extension and the Repast software from the ArcMap interface.  The extension installation provided tutorial projects and data which I was able to work through.

The Agent Analyst website also provided a two part tutorial described below for the Schelling segregation model.  A video of my completed tutorial model running within ArcGIS can be found here:  <http://mypage.iu.edu/~jppeters/AgentAnalyst_SegregationModel.swf>

*From the* [*Agent Analyst website*](http://www.institute.redlands.edu/AgentAnalyst/tutorials.aspx): <http://www.institute.redlands.edu/AgentAnalyst/tutorials.aspx>

[Schelling GIS](http://www.institute.redlands.edu/AgentAnalyst/Tutorials/SchellingTutorial.zip) - *Arika Ligmann-Zielinska, Ph.D. Associate, San Diego State University* This two part tutorial, developed for an Advanced GIS course at SDSU, provides an excellent introduction to Agent Analyst using the Schelling sample model and data included with the Agent Analyst installation. <http://www.institute.redlands.edu/AgentAnalyst/Tutorials/SchellingTutorial.zip>

As I ran the model multiple times, I thought the existing model could be significantly enhanced.  For example, if the density parameter was initially set to a low number there would be very few “moves” because the tolerance threshold was not being met due to the vacant neighborhoods.

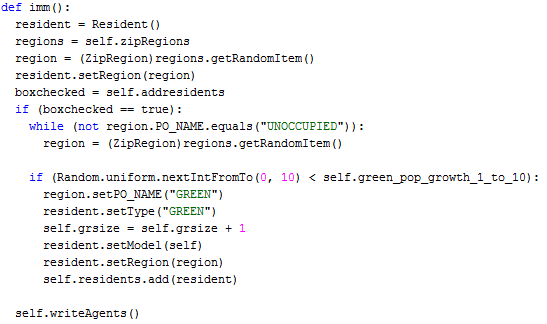
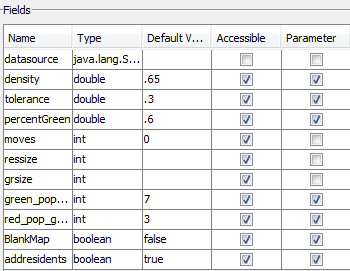
After completing the Agent Analyst tutorials and getting a feel for the extension I decided to enhance the segregation model and to use a different polygon shapefile to represent “neighborhoods” within the model.

This step was not as simple as using a different shapefile as the source file.  The model is dependent upon an auxiliary text file which defines all of the “neighbors” for each polygon.  The example below illustrates the data preparation that is needed for each polygon to be used in the model.  As outlined below the polygon 60076 has 7 neighbors each as a comma separated value.

60076: 60077,46394,60202,60645,60712,60659,60610

Fortunately, existing tools within ArcGIS allow a text output of adjacent polygons to a .csv file which I was then able to convert to a text file.  I decided to use the shapefile of the new Indiana General Assembly House districts that came out of the redistricting process that was recently completed.  There are 100 districts which made checking various components of the model easier to analyze such as agent percentages and totals.  In earlier iterations I had also prepared parcel data to use in the model, but by using the Indiana House districts, it visually conveys that it is a local dataset that is being used in the model.

Upon updating the model to use local data, I wanted to add additional parameters for running the model.  My idea was to add variables to the model to simulate population growth parameters for both the Red and Green populations.  These parameters would control how often a new agent was spawned in each of the populations.  This is intended to show how this type of segregation (and the total amount of agent moves) increase as the population increases.

The model started with 4 Fields, of which, 3 were model parameters (Percent Green, Density, and Tolerance).  The fourth field was the source of the shapefile, which I also edited.  I added 7 new fields to the model.  As previously mentioned, Red population growth and Green population growth are now parameters in the model.  At model setup the user can specify the probability of growth per tick for each population.

To realize this spawning (immigration) of agents based on a probability parameter I created two new actions that would be controlled independently by 2 model parameters available at the start screen of the model. The example to the right is the action controlling the Green population growth. These two population growth parameters are controlled by, and only active within the model, if the user wants to simulate population growth within the model. This option is enabled by default via the “addresidents” checkbox.

I have also included a new “BlankMap” parameter within the enhanced model as a Boolean checkbox. This checkbox allows the user to specify whether to start with a Blank Map.  This will override density and percentage green parameters if checked and start the model with all neighborhoods being vacant.  Residents will then be added to the model based solely on the population growth parameters specified.

The final three new fields added to the model are primarily for the graph.  In order to display running totals of each of the populations as well as a count of moves per turn, fields were added to the model with appropriate updates at each tick of the model.  To ensure that the total moves per turn was reset to “0” at the start of each turn, code was added to the “UpdateGISDisplay” series which is scheduled to take place each tick.

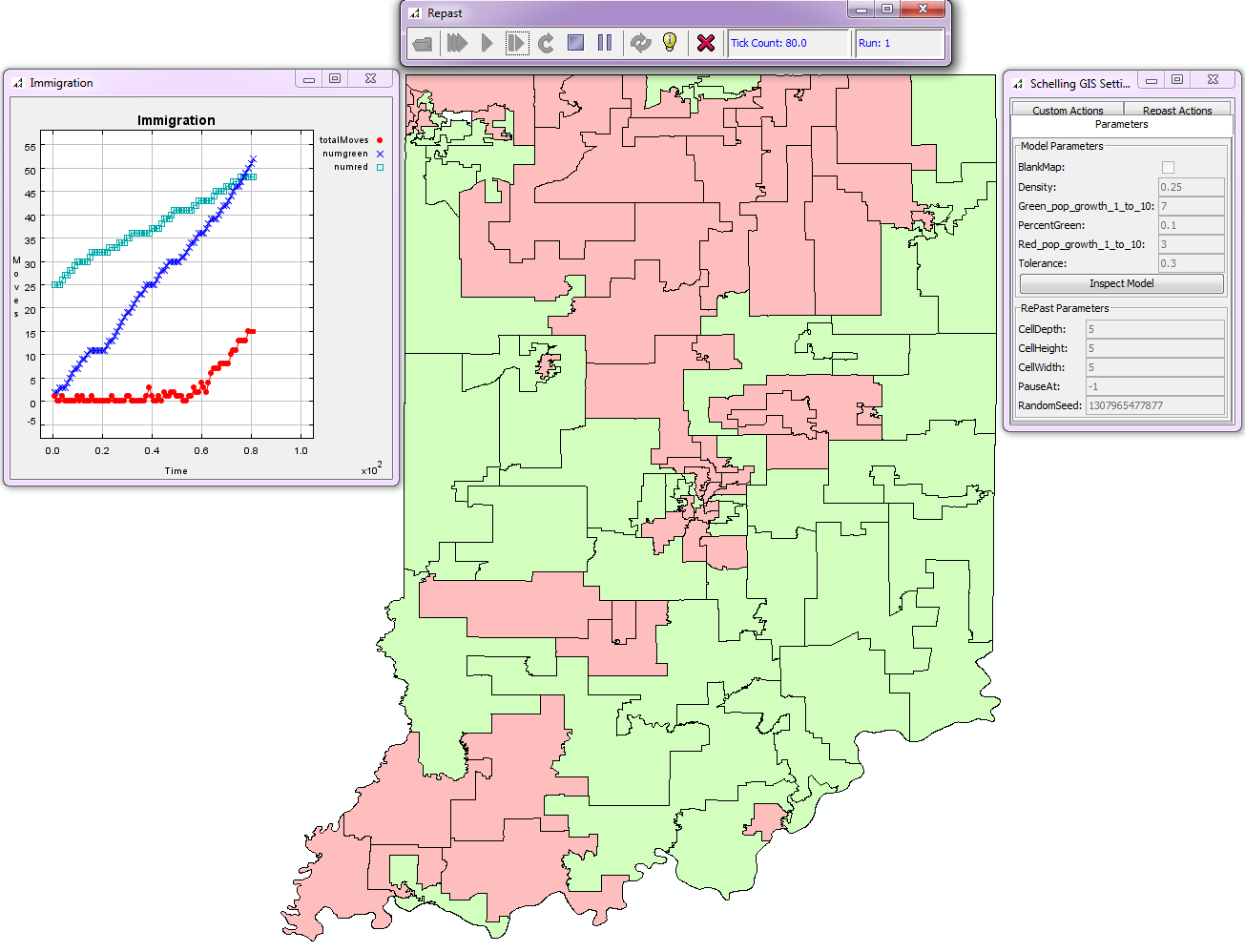


Figure 1. In the model above although the green population only started with 2 residents, at tick 80 there were 52 green residents. The red population started with 25 residents, but with a lower population growth parameter, the red population ended with 48 residents. There was very little movement per tick due to segregation until approximately tick 60 when the area was approximately 75% full of residents.

Although this model focuses primarily on spawning new agents based on population growth parameters, as to simulate immigration, only a few changes would be needed to use the model to elicit a new model to address issues of emigration such as the “White Flight” concept.  This model could easily be changed to have one population group leaving or agents “dying” instead of spawning at given intervals if a tolerance threshold is reached.

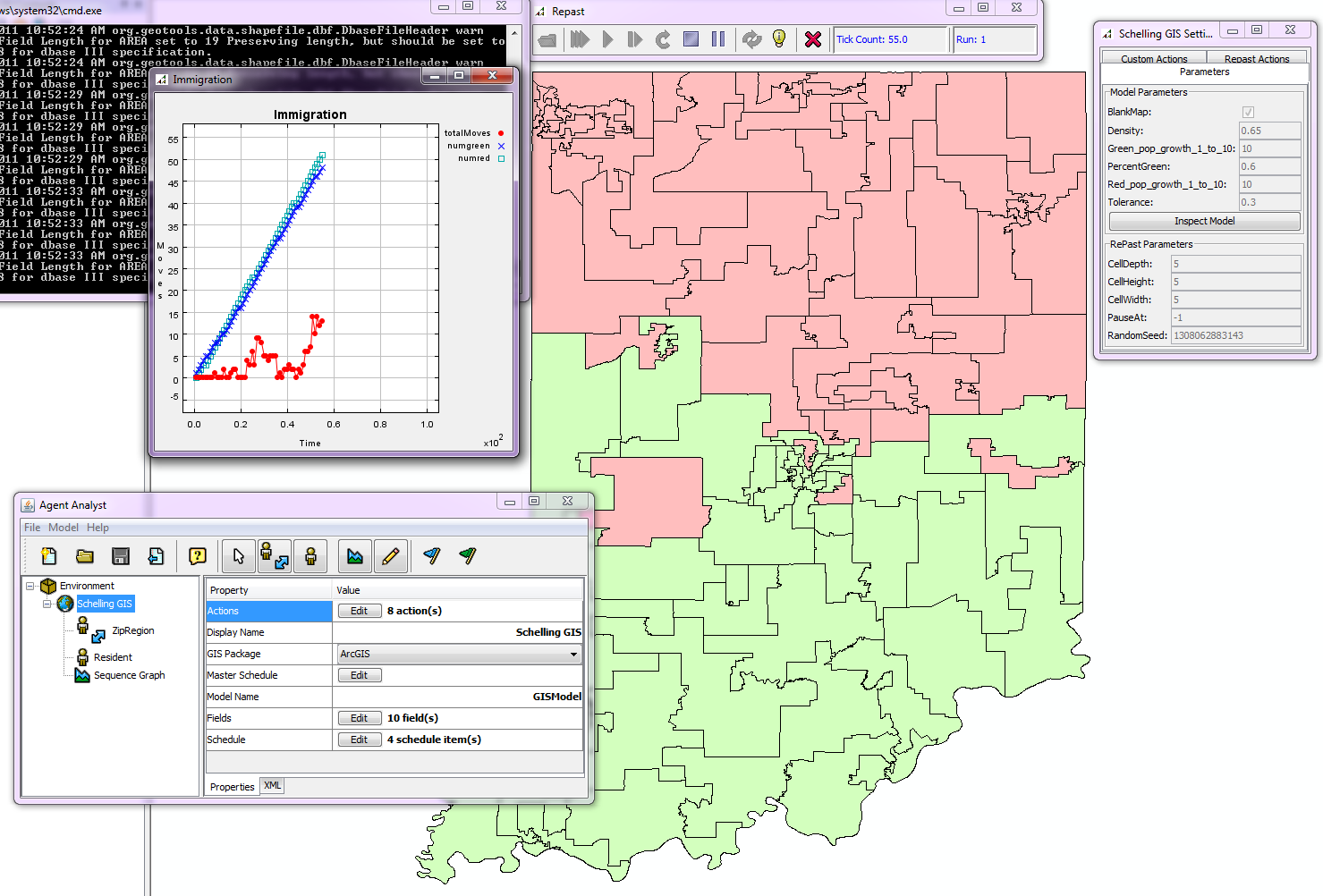


Figure 2. In the model above both populations were given the same probability of growth and started with a blank map.

Screen capture images of the model parameters and the Repast model design interface:

Download for shapefile used:

<http://inmap.indiana.edu/downloads/Indiana_House_117GA_IED.zip>

Zip file containing project and neighbors auxiliary text file:

<http://ella.slis.indiana.edu/~jppeters/S603ABM.zip>

Screen capture video of model running:

<http://ella.slis.indiana.edu/~jppeters/S603ABM.swf>

